

WHAT IS CLAIMED IS:

1 1. A processor, comprises:

2 an execution unit for executing multiple context threads, said execution unit

3 comprises:

4 an arithmetic logic unit to process operands for executing threads;

5 a multiplexor;

6 control logic to control the operation of the arithmetic logic unit and the multiplexor;

7 an immediate data bus coupled from the output of the control logic to an input of the
8 multiplexor to provide immediate data to the arithmetic logic unit through the multiplexor;

9 a general purpose register coupled to the input of the multiplexor to provide register
10 operand data to the arithmetic logic unit through the multiplexor; and

11 a read transfer register coupled to the input of the multiplexor to provide operand data
12 from a memory device;

13 wherein execution of a register instruction causes data from one or more input
14 sources connected to the multiplexor to be transferred through the multiplexor into the
15 arithmetic logic unit and causes data to be transferred through the arithmetic logic unit to one
16 of the registers.

1 2. The execution unit of claim 1 wherein the register instruction includes a bit

2 mask specifying which one or more bytes are affected and causing the bytes specified by the
3 bit mask to be loaded from the immediate data bus and the bytes not specified to be loaded
4 from the read transfer register.

1 3. The execution unit of claim 1 wherein the register instruction includes a bit

2 mask specifying which one or more bytes are affected and causing the bytes specified by the
3 bit mask to be loaded from the immediate data bus and the bytes not specified to be loaded
4 from the general purpose register.

1 4. The execution unit of claim 1 further comprising:

2 a bypass bus coupled from the output of the arithmetic logic unit to an input of the
3 multiplexor; and

4 control logic to control the execution of a series of pipelined instructions wherein
5 each pipelined instruction may specify a read part and a write part, where the read part of one
6 pipelined instruction specifies a read address that is the same as a write address of the write
7 part of another pipelined instruction causing the data being written by the write part to be
8 available to the read part in the same processor cycle.

1 5. A method for executing multiple context threads comprises:
2 processing operands for an executing thread through a multiplexor and an arithmetic
3 logic unit;

4 operating control logic connected to the arithmetic logic unit and the multiplexor;
5 providing immediate data on an immediate data bus coupled from the output of the
6 control logic to an input of the multiplexor;

7 providing operand data to the arithmetic logic unit from a general purpose register
8 coupled to the input of the multiplexor;

9 providing operand data from a memory device through a read transfer register
10 coupled to the input of the multiplexor; and

11 executing a register instruction to cause data from one or more input sources
12 connected to the multiplexor to be transferred through the multiplexor into the arithmetic
13 logic unit and to cause data to be transferred through the arithmetic logic unit to one of the
14 registers.

1 6. The method of claim 5 wherein executing a register instruction further
2 comprises:

3 including a bit mask; and
4 causing the bytes specified by the bit mask to be transferred through the multiplexor
5 from the immediate data bus and the bytes not specified to be transferred through the
6 multiplexor from the transfer register.

1 7. The method of claim 5 wherein executing a register instruction further
2 comprises:

3 including a bit mask; and
4 causing the bytes specified by the bit mask to be transferred through the multiplexor
5 from the immediate data bus and the bytes not specified to be transferred through the
6 multiplexor from the general purpose register.

1 8. A computer program product residing on a computer readable medium
2 causing a processor to perform a function comprises instructions causing the processor to:
3 perform a register operation which specifies a bit mask corresponding to one or more
4 bytes of data to cause the bytes specified by the bit mask to be loaded from an immediate
5 data bus and the bytes not specified to be loaded from a register data bus.